

Advanced Computer Networks

Internal routing - distance vector protocols

*Prof. Andrzej Duda
duda@imag.fr*

`http://duda.imag.fr`

Contents

- Principles of internal routing
- Distance vector (Bellman-Ford)
 - principles
 - case of link failures
 - count to infinity
 - split horizon
- RIP
- RIP v2
- IGRP

Routing algorithms

- Problem
 - find the **best** route to a destination
- What does it mean the best?
 - metric to measure how a route is good
 - hops
 - link capacity
 - performance measures: link load, delay
 - cost
- Graph optimization - Shortest Path
 - find the shortest path in a graph
 - shortest in the sense of a metric

Main algorithms

- Distance vector (Bellman-Ford)
 - routers only know their local state
 - link metric and neighbor estimates
 - internal routing protocols (RIP, IGRP)
- Link state
 - knowledge of the global state
 - metrics of all links
 - global optimization (Shortest Path First - Dijkstra)
 - internal routing protocols (OSPF, PNNI (ATM))
- Path vector
 - knowledge of the global state
 - path: sequence of AS with attributes
 - global optimization and policy routing
 - external routing protocols (BGP)

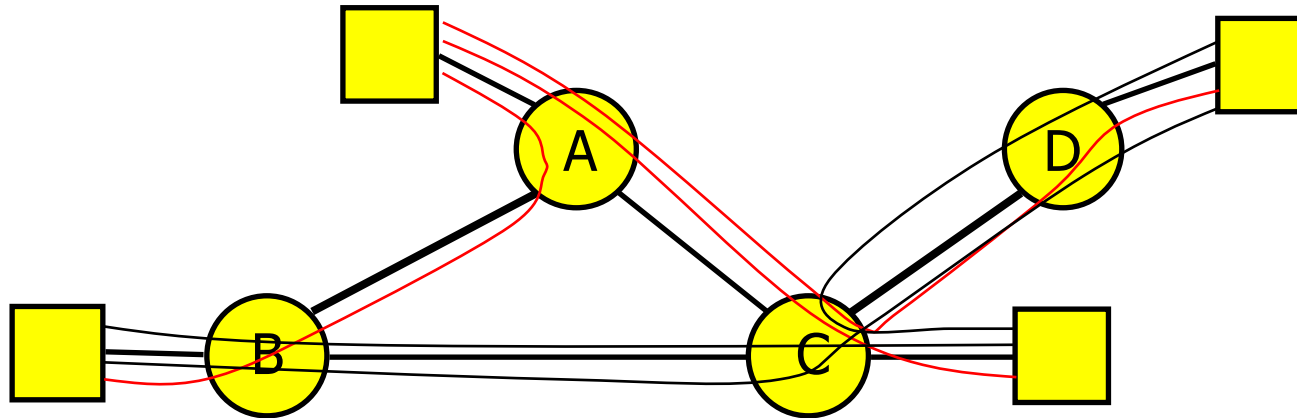
Routing protocols

	Internet	ISO
IGP	distance vector: RIP, RIP v2, IGRP	
	link state: OSPF dual: EIGRP	IS-IS
EGP	EGP (obsolete) BGP	IDRP
host	ICMP Redirect	IS-ES

Metrics

- Static - do not depend on the network state
 - number of hops
 - link capacity and static delay
 - cost
- Dynamic - depend on the network state
 - link load
 - current delay

Traffic matrix

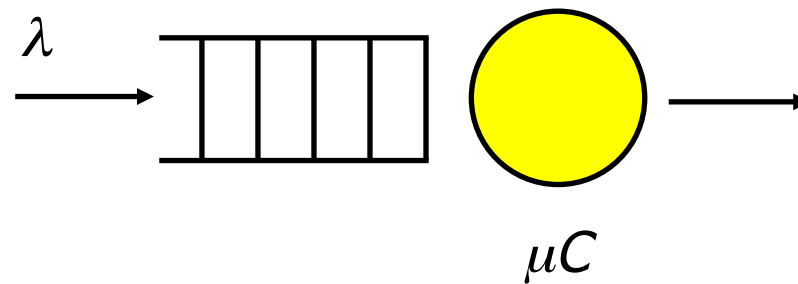


	A	B	C	D
A	0	8	2	3
B		0	6	4
C			0	1
D				0

	A	B	C	D
A		AB	AC	ACD
B			BC	BCD
C				CD
D				

Traffic

- Link model
 - queueing system M/M/1
 - exponentially distributed service and interarrival times



$$T = \frac{1}{\mu C - \lambda}$$

Delay

- Parameters

- 1 Mb/s and 0.5 Mb/s links
- mean packet length $1/\mu$ - 5 Kbytes (40 000) bits
- transmission time on 1 Mb/s link: 40 ms
- transmission time on 0.5 Mb/s link: 80 ms

	λ pq/s	C Mb/s	μC pq/s	T
AB	8	1	25	58 ms
AC	5	0.5	12.5	133 ms
BC	10	0.5	12.5	400 ms
CD	8	1	25	58 ms

Flooding

- Simple and robust routing
 - no need for routing tables
 - each packet duplicated on each outgoing link
 - packet duplication
 - duplicated packets destroyed at destination
 - robust - tolerates link or router failures
 - optimal in some sense
 - the first packet has found the shortest path to the destination
 - cannot be compared to the shortest path calculated by Link State - no packet duplication
- Problem
 - increased load due to packet duplication
- Used in OSPF to distribute link state information and in ad hoc routing protocols (AODV, OLSR)

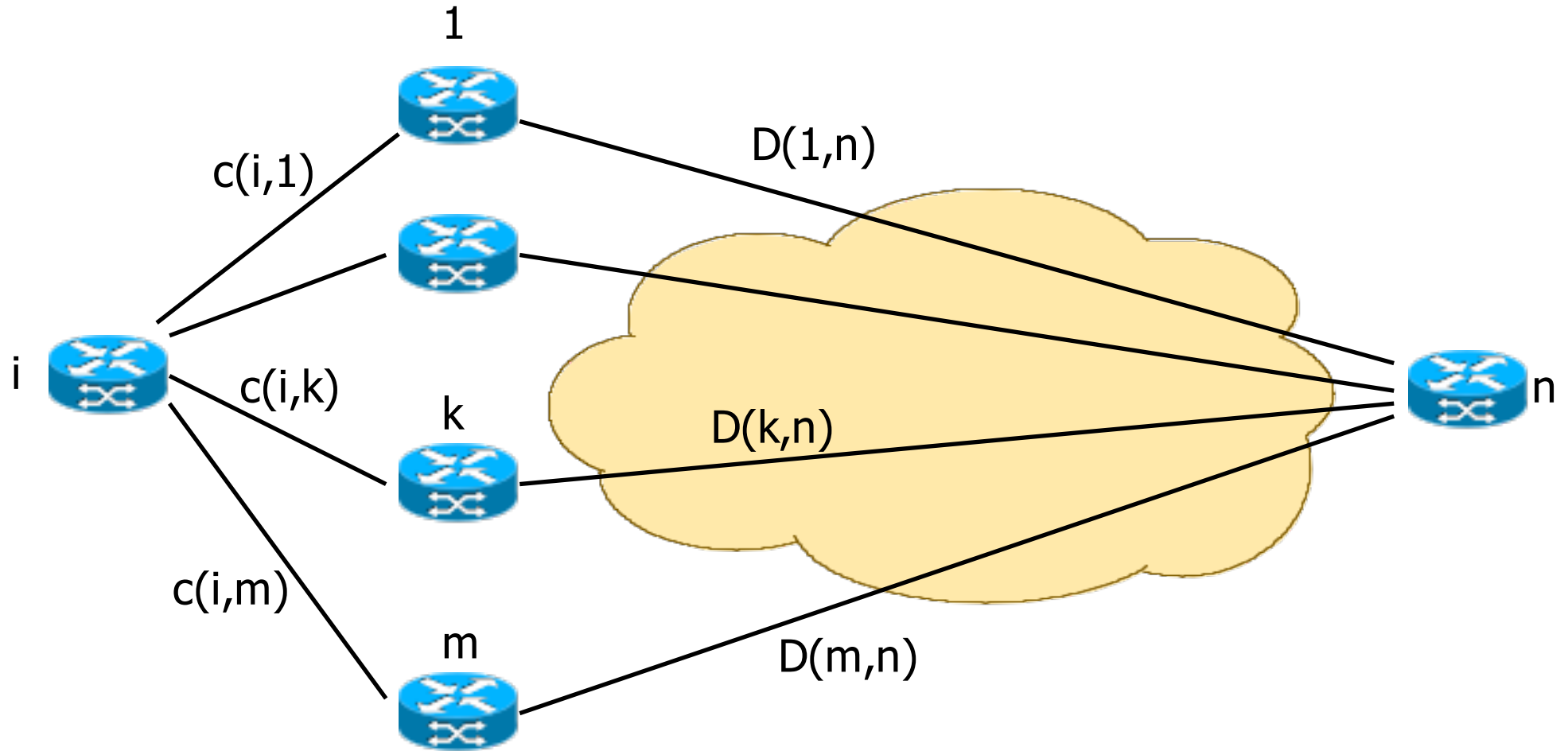
Distance vector

- Dynamic routing based on distributed estimation of the distance to the destination
 - uses the distributed algorithm by Bellman-Ford (dynamic programming)
 - each router receives aggregated information from its neighbors
 - estimates the local cost to its neighbors
 - computes the best routes
 - no global network states
- Distance
 - number of hops
 - delay

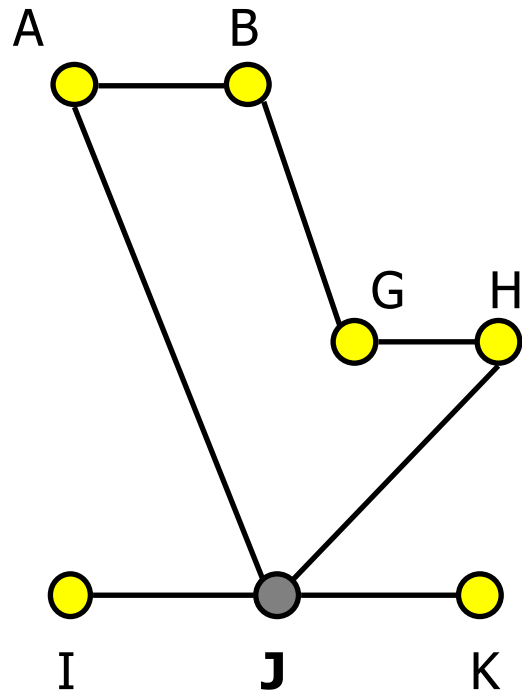
Bellman-Ford algorithm

- Bellman-Ford algorithm
 - node i knows cost $c(i,k)$ to its immediate neighbours ($+\infty$ for most values of k)
 - distance $D(i,n)$ is given by: $D(i,n) = \min_k (c(i,k) + D(k,n))$
 - in the worst case, convergence after $N-1$ iterations
- Distributed Bellman-Ford algorithm
 - initially: $D(i,n) = 0$ if i directly connected to n and $D(i,n) = +\infty$ otherwise
 - node i receives from neighbour k latest values of $D(k,n)$ for all n (distance vector)
 - node i computes the best estimates
$$D(i,n) = \min_k (c(i,k) + D(k,n))$$

Bellman-Ford algorithm



Example of Bellman-Ford



	A	I	H	K
A	0	24	20	21
B	12	36	31	28
G	18	31	6	31
H	17	20	0	19
I	21	0	14	22
J	9	11	7	10
K	24	22	22	0
J:	8	10	12	6

Table of J

8	A
20	A
18	H
12	H
10	I
0	-
6	K

computation of G : $18+8=26$, $31+10=41$, $6+12=18$, $6+31=37$
 → choice of 18, H

Distance vector example

- Simple network
 - routers connected by links
 - destinations = subnetworks connected to routers
 - symmetric links
 - cost = number of hops

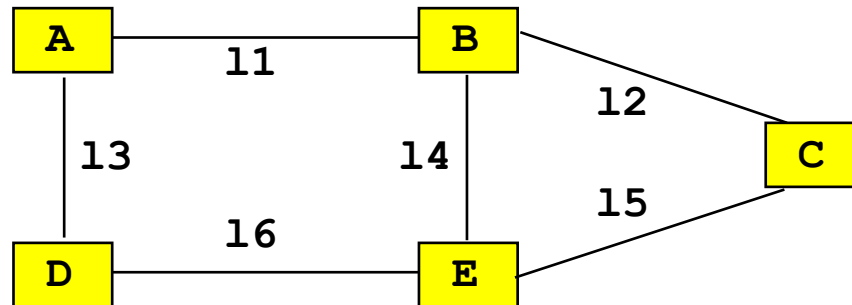
Initialization

A

dest	link	cost
A	local	0

B

dest	link	cost
B	local	0



D

dest	link	cost
D	local	0

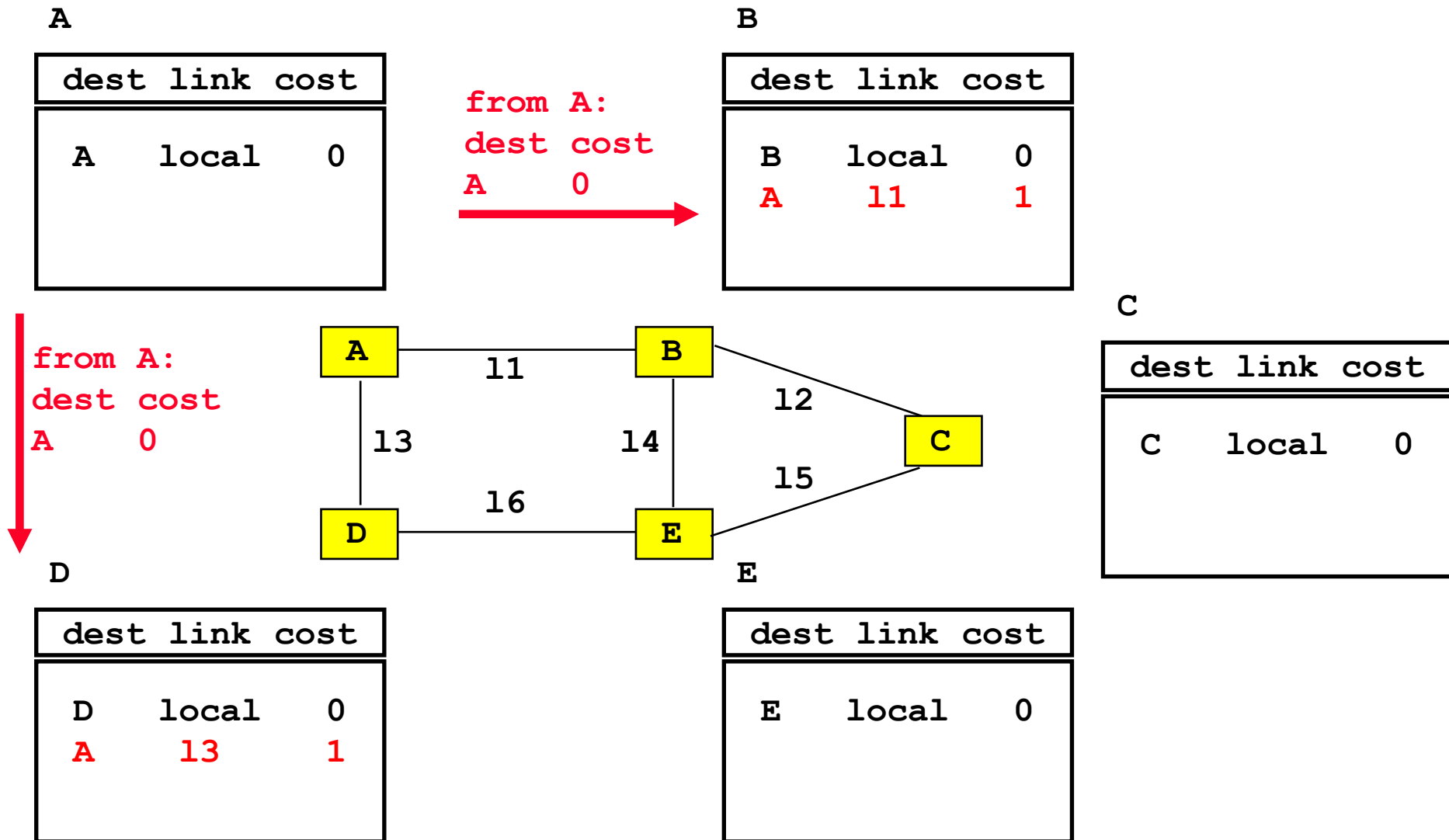
E

dest	link	cost
E	local	0

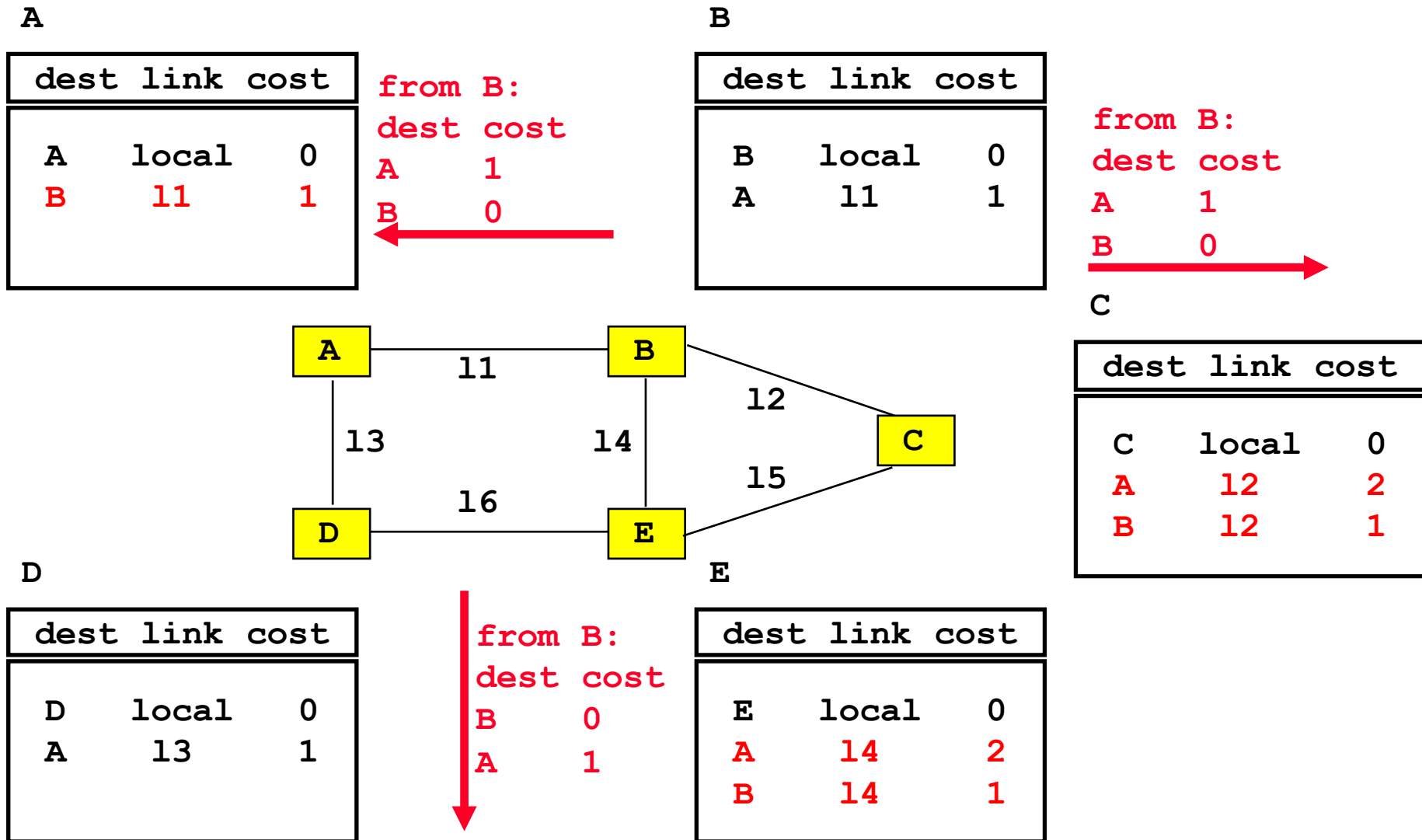
C

dest	link	cost
C	local	0

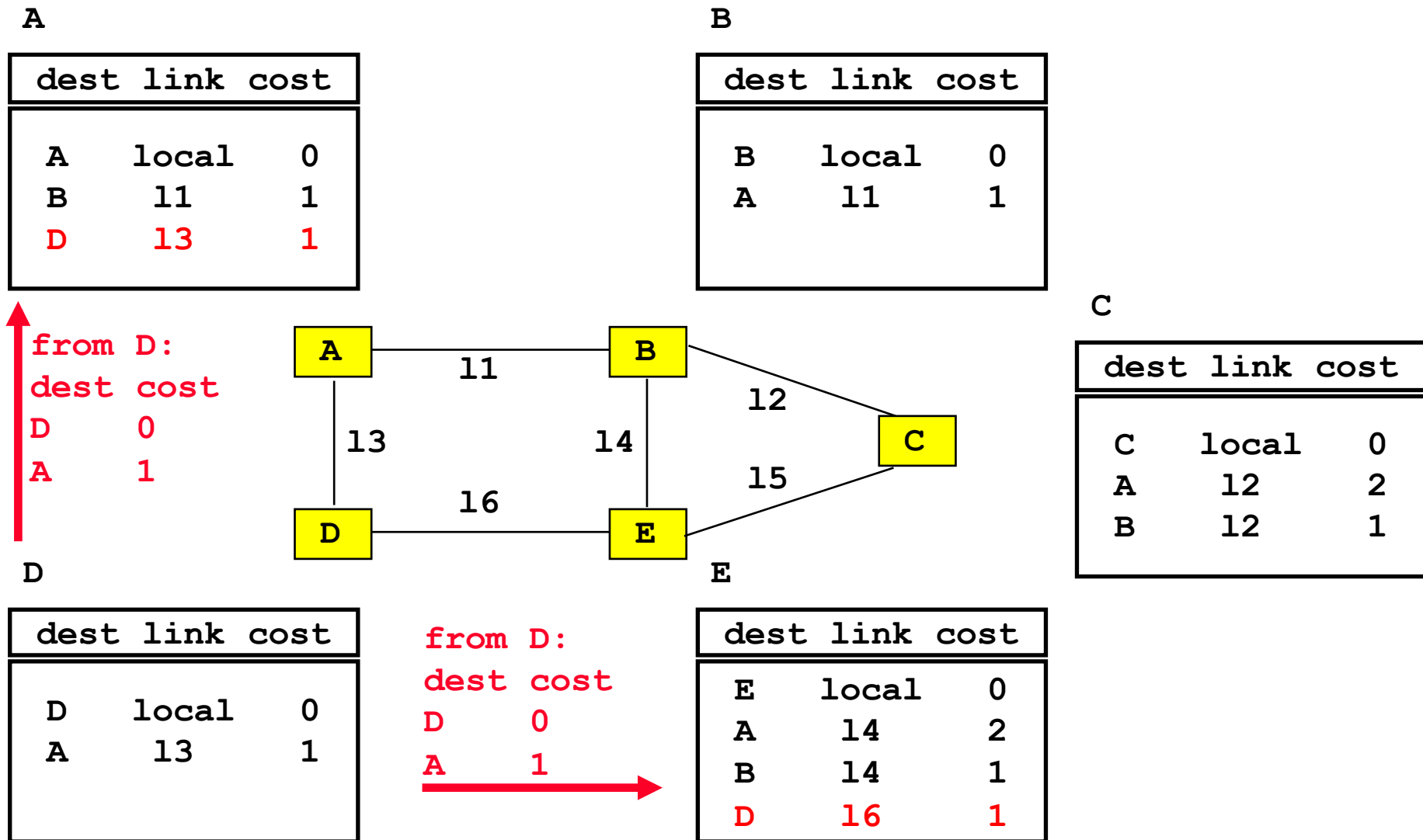
Distance vector announcement



Distance vector announcement



Distance vector announcement



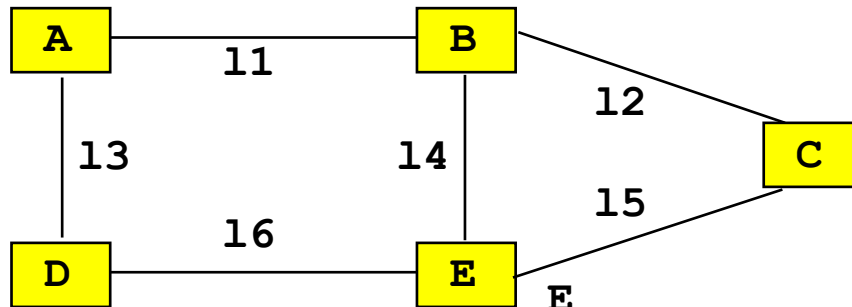
Final

A

dest	link	cost
A	local	0
B	11	1
D	13	1
C	11	2
E	11	2

B

dest	link	cost
B	local	0
A	11	1
C	12	1
E	14	1
D	11	2



C

dest	link	cost
C	local	0
A	12	2
B	12	1
D	15	2
E	15	1

D

dest	link	cost
D	local	0
A	13	1
B	13	2
C	16	2
E	16	1

dest	link	cost
E	local	0
A	14	2
B	14	1
D	16	1
C	15	1

Link failure

A

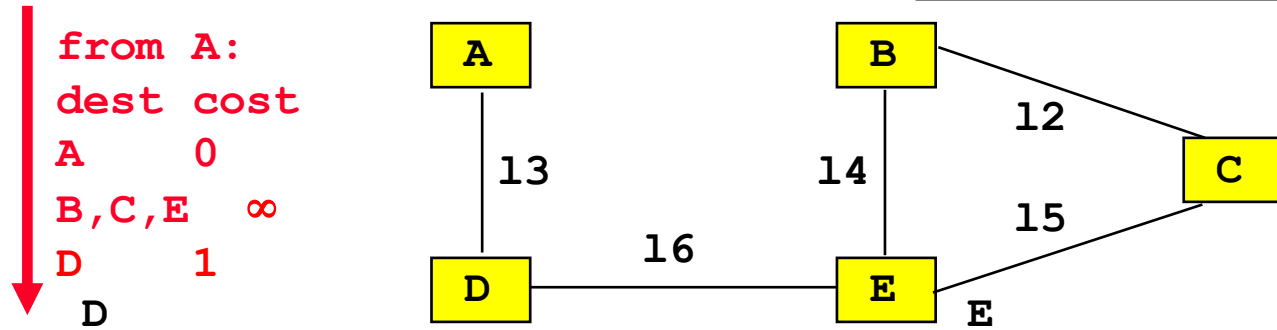
dest	link	cost
A	local	0
B	11	∞
D	13	1
C	11	∞
E	11	∞

B

dest	link	cost
B	local	0
A	11	∞
C	12	1
E	14	1
D	11	∞

C

dest	link	cost
C	local	0
A	12	∞
B	12	1
D	15	2
E	15	1



from A:
dest cost
A 0
B,C,E ∞
D 1

dest	link	cost
D	local	0
A	13	1
B	13	∞
C	16	2
E	16	1

dest	link	cost
E	local	0
A	14	∞
B	14	1
D	16	1
C	15	1

Link failure

A

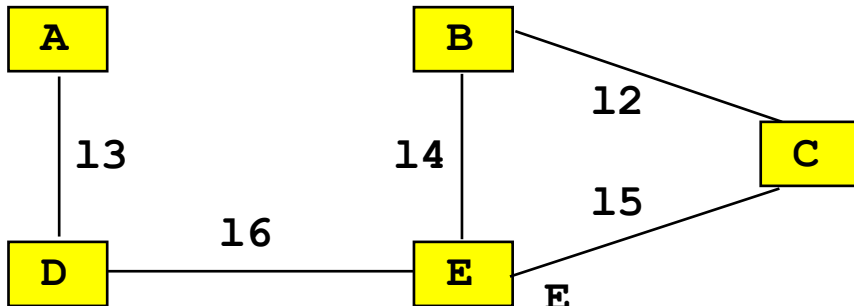
dest	link	cost
A	local	0
B	11	∞
D	13	1
C	11	∞
E	11	∞

B

dest	link	cost
B	local	0
A	11	∞
C	12	1
E	14	1
D	11	∞

C

dest	link	cost
C	local	0
A	12	∞
B	12	1
D	15	2
E	15	1



D

dest	link	cost
D	local	0
A	13	1
B	16	2
C	16	2
E	16	1

from B:
 dest cost
 E 0
 A ∞
 B,D,C 1



dest	link	cost
E	local	0
A	14	∞
B	14	1
D	16	1
C	15	1

Final state after failure

A

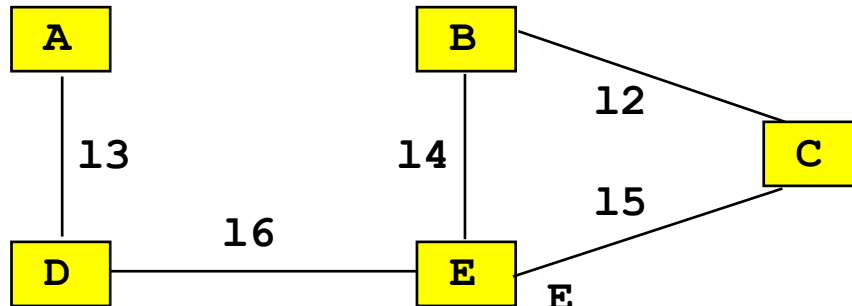
dest	link	cost
A	local	0
B	13	3
D	13	1
C	13	3
E	13	2

B

dest	link	cost
B	local	0
A	14	3
C	12	1
E	14	1
D	14	2

C

dest	link	cost
C	local	0
A	15	3
B	12	1
D	15	2
E	15	1



D

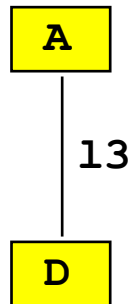
dest	link	cost
D	local	0
A	13	1
B	16	2
C	16	2
E	16	1

dest	link	cost
E	local	0
A	16	2
B	14	1
D	16	1
C	15	1

Equal link costs - link failures

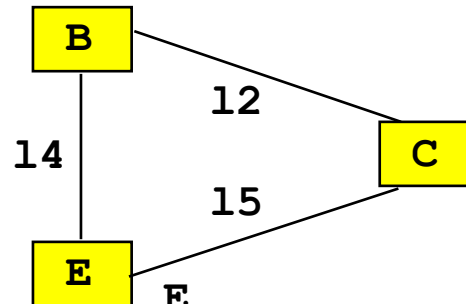
A

dest	link	cost
A	local	0
B	13	3
D	13	1
C	13	3
E	13	2



B

dest	link	cost
B	local	0
A	14	3
C	12	1
E	14	1
D	14	2



C

dest	link	cost
C	local	0
A	15	3
B	12	1
D	15	2
E	15	1

D

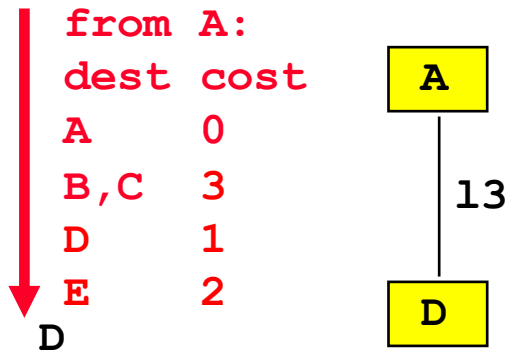
dest	link	cost
D	local	0
A	13	1
B	16	∞
C	16	∞
E	16	∞

dest	link	cost
E	local	0
A	16	2
B	14	1
D	16	1
C	15	1

Counting to infinity

A

dest	link	cost
A	local	0
B	13	3
D	13	1
C	13	3
E	13	2



dest	link	cost
D	local	0
A	13	1
B	13	4
C	13	4
E	13	3

- Loop between A and D
- Exchange of routes, costs increase by 2 each cycle
- Convergence to a stable state
 - ∞ = large number
 - e.g. RIP: ∞ = 16

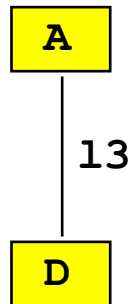
Split horizon

- Minimize the effects of bouncing and counting to infinity
- Rule
 - if A routes packets to X via B, it does not announce this route to B

Example of split horizon

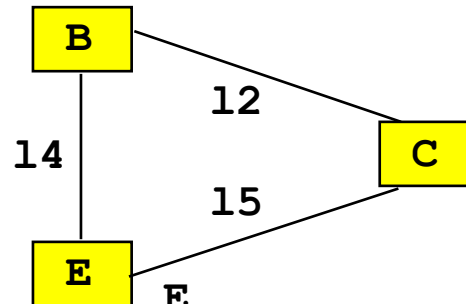
A

dest	link	cost
A	local	0
B	13	3
D	13	1
C	13	3
E	13	2



B

dest	link	cost
B	local	0
A	14	3
C	12	1
E	14	1
D	14	2



C

dest	link	cost
C	local	0
A	15	3
B	12	1
D	15	2
E	15	1

D

dest	link	cost
D	local	0
A	13	1
B	16	∞
C	16	∞
E	16	∞

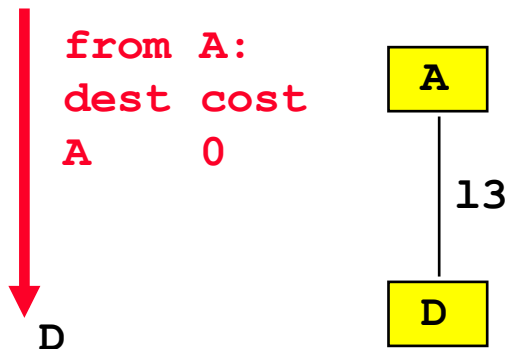
dest	link	cost
E	local	0
A	16	2
B	14	1
D	16	1
C	15	1

Split horizon

A

dest	link	cost
A	local	0
B	13	3
D	13	1
C	13	3
E	13	2

- Split horizon cuts the process of counting to infinity



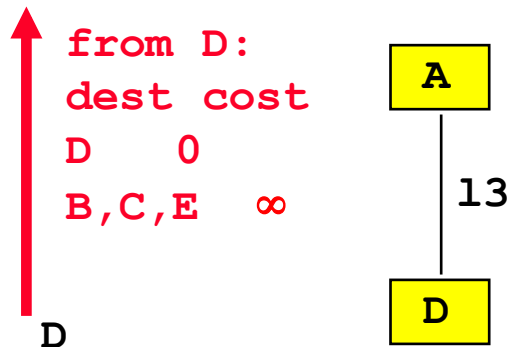
dest	link	cost
D	local	0
A	13	1
B	16	∞
C	16	∞
E	16	∞

Split horizon

A

dest	link	cost
A	local	0
B	13	∞
D	13	1
C	13	∞
E	13	∞

- Split horizon cuts the process of counting to infinity



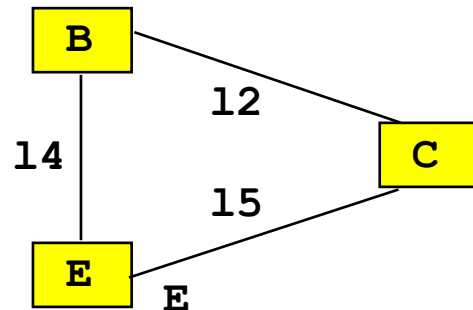
dest	link	cost
D	local	0
A	13	1
B	16	∞
C	16	∞
E	16	∞

Split horizon may fail

↑ from E:

dest	cost
E	0
A	∞
C	1
D	∞

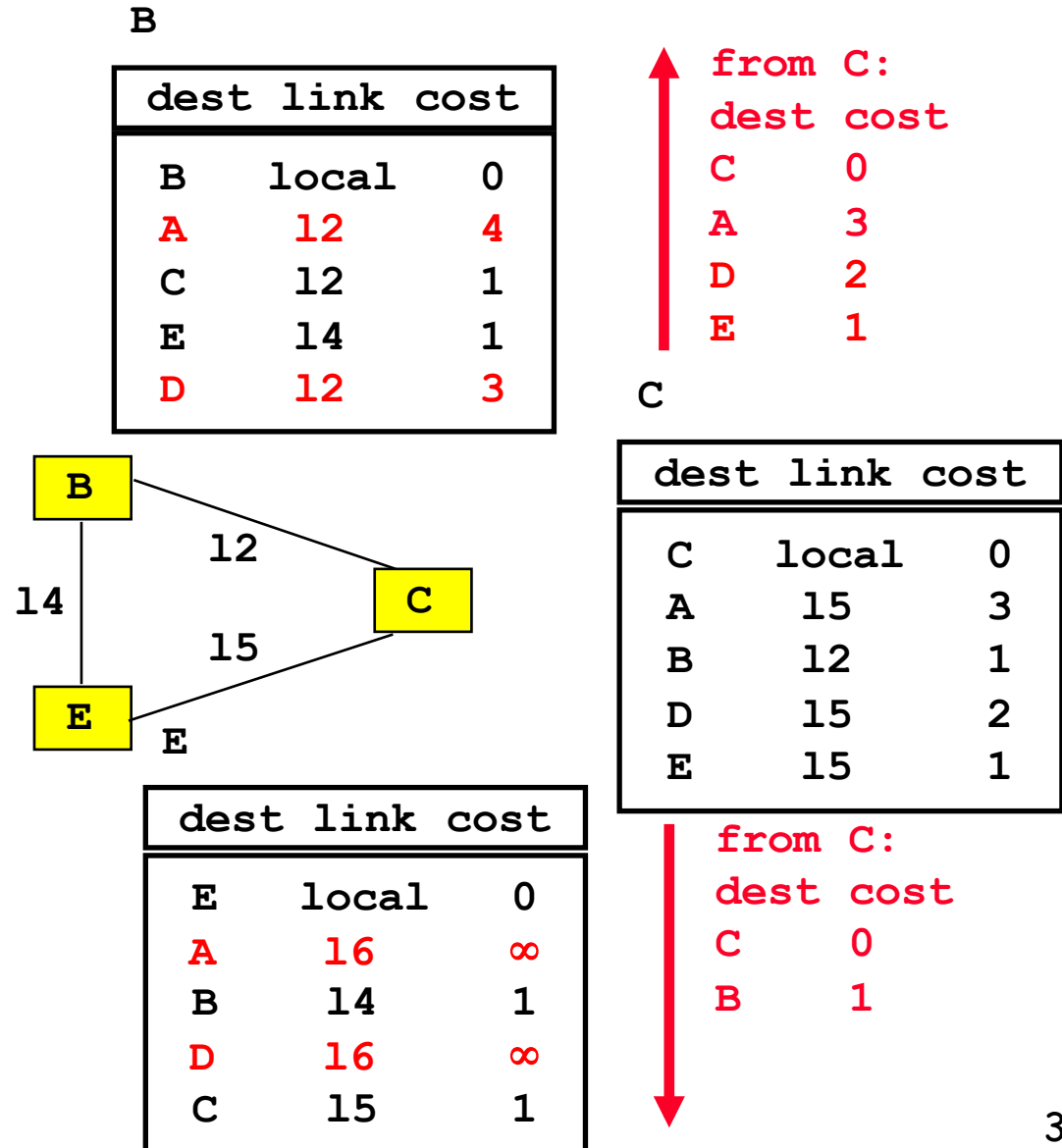
B		
dest	link	cost
B	local	0
A	14	∞
C	12	1
E	14	1
D	14	∞



C		
dest	link	cost
C	local	0
A	15	3
B	12	1
D	15	2
E	15	1

E		
dest	link	cost
E	local	0
A	16	∞
B	14	1
D	16	∞
C	15	1

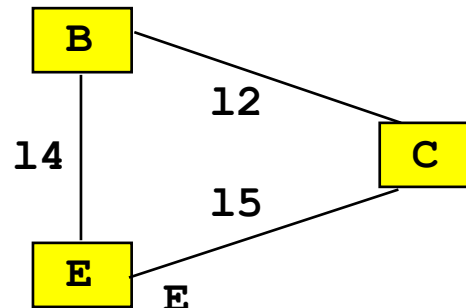
Split horizon may fail



Split horizon may fail

B

dest	link	cost
B	local	0
A	12	4
C	12	1
E	14	1
D	12	3



C

dest	link	cost
C	local	0
A	15	3
B	12	1
D	15	2
E	15	1

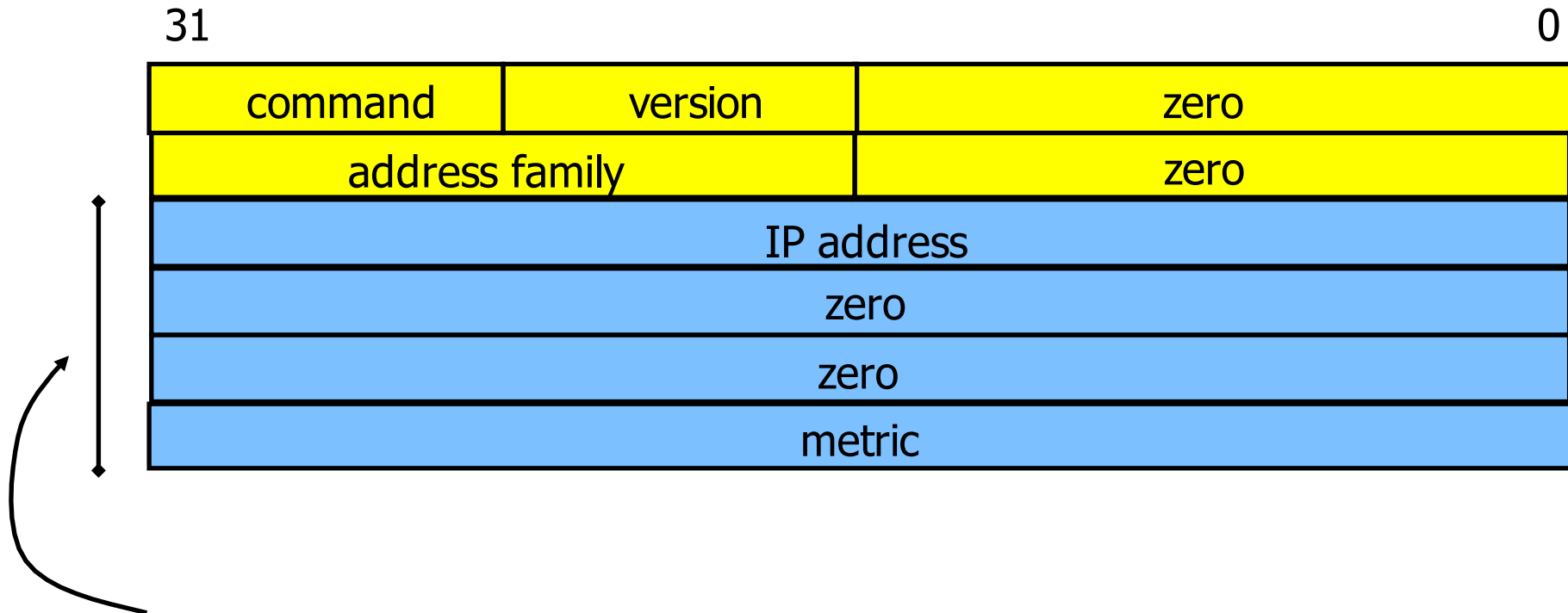
from B:
 ↓
 dest cost
 A 4
 B 0
 C 1
 D 3

dest	link	cost
E	local	0
A	14	5
B	14	1
D	14	4
C	15	1

RIP v1

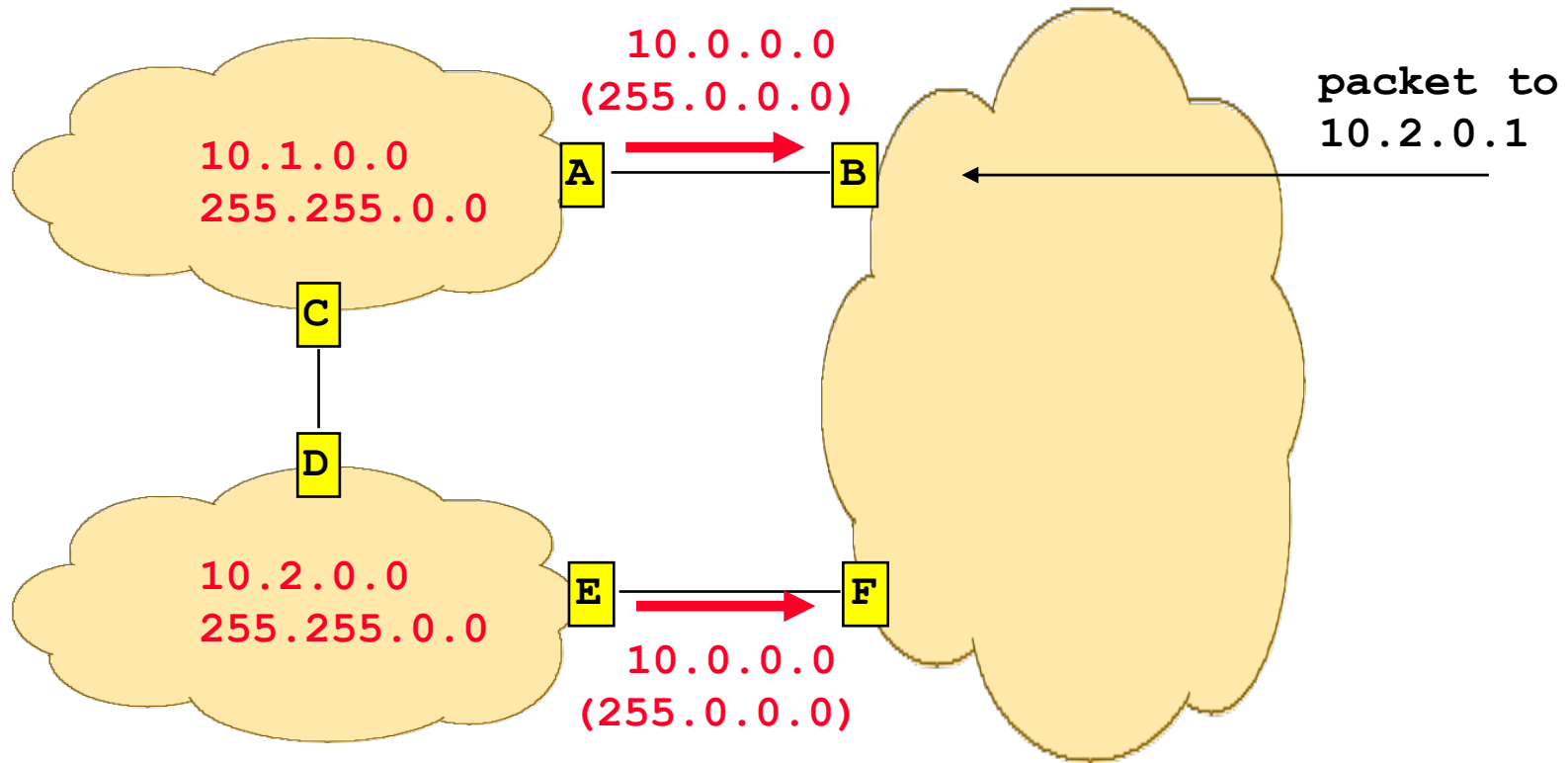
- Distance vector protocol
- Metric - hops
- Network span limited to 15
 - $\infty = 16$
- Split horizon
- Destination network identified by IP address
 - no prefix/subnet information - derived from address class
- Encapsulated as UDP packets, port 520
- Largely implemented (**routed** on Unix)
- Broadcast every 30 seconds or when update detected
- Route not announced during 3 minutes
 - cost becomes ∞

Message format



- May be repeated 25 times
- Command
 - **REQUEST** - 1 (sent at boot to initialize)
 - **RESPONSE** - 2 (broadcast each 30 sec)

Missing netmask

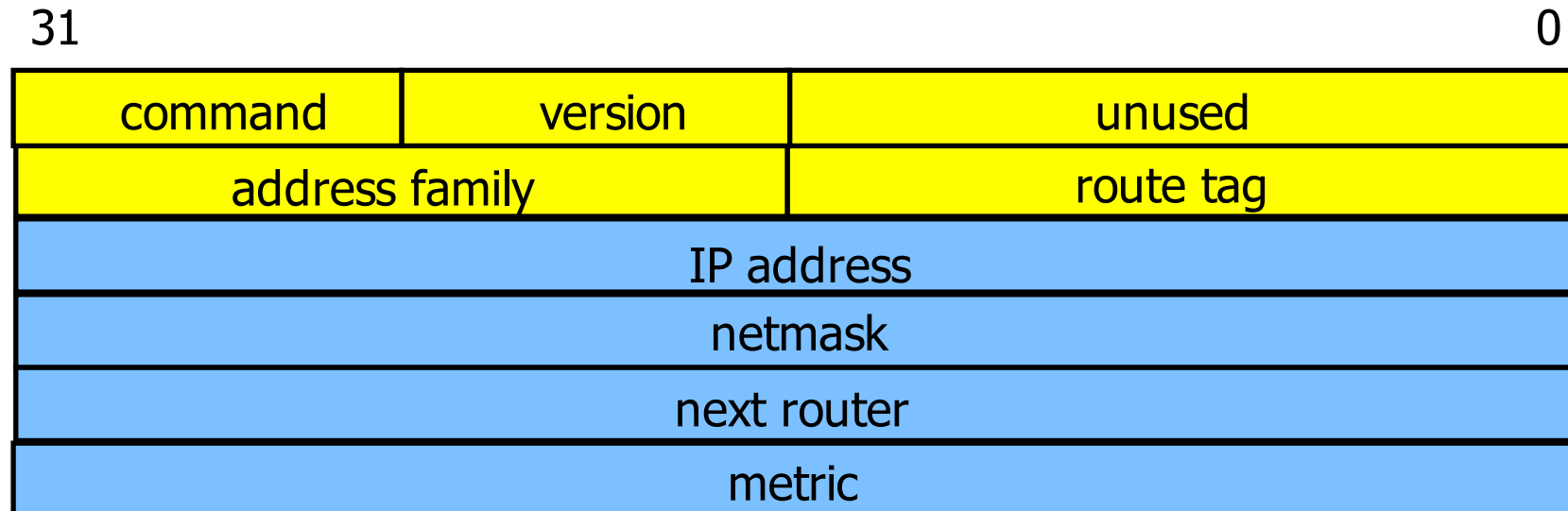


- A and E can forward to 10.0.0.0
- Packet to 10.2.0.1 can go through F or B
 - if sent to B, it goes through A and C
- If link C-D broken, no route to destination

RIP v2 (RFC 2453)

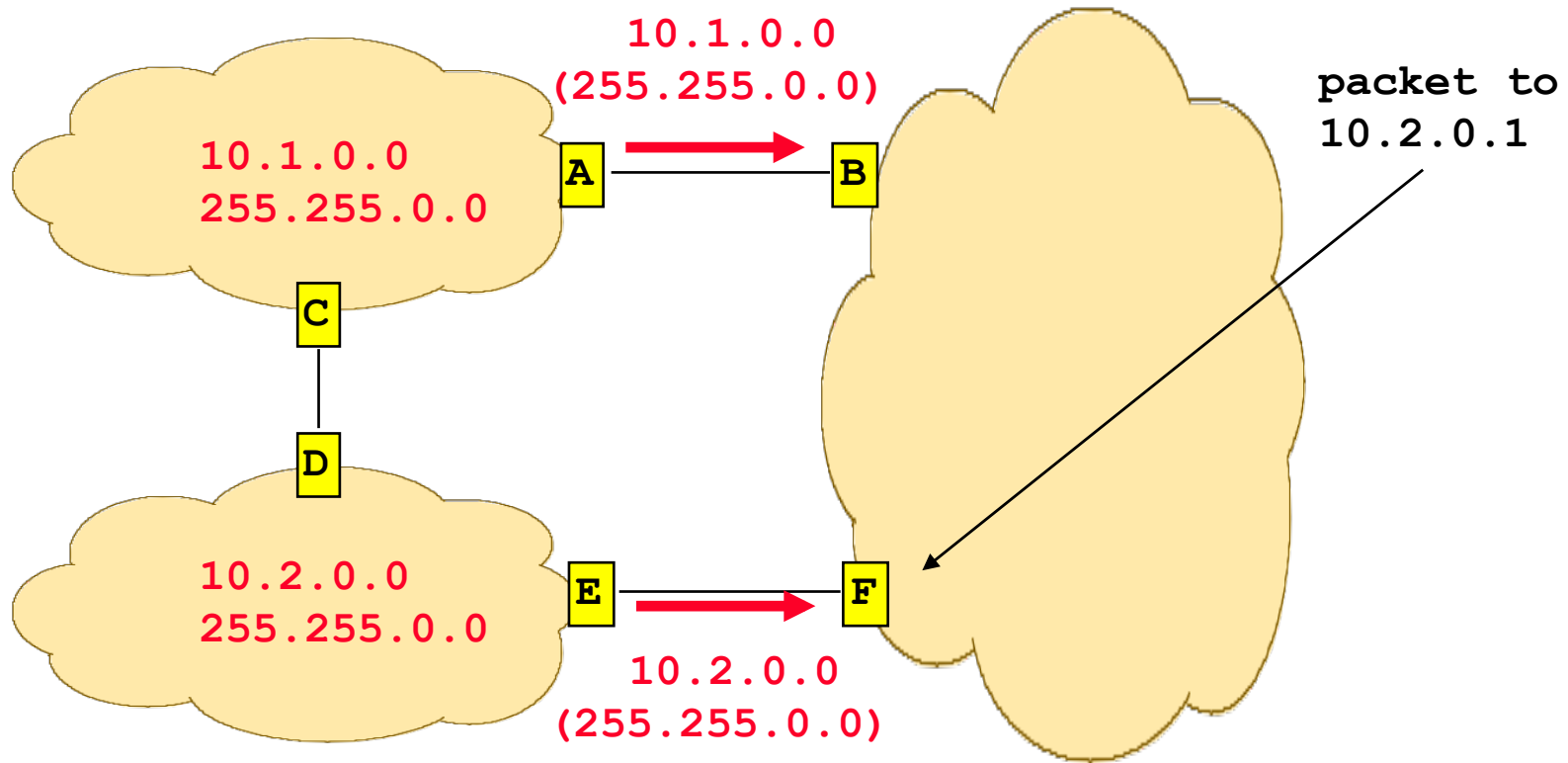
- Subnetworks
 - take into account CIDR prefixes and netmasks
- Authentication
- Multicast
 - 224.0.0.9 mapped to MAC 01-00-5E-00-00-09
 - on LAN only, no need for IGMP

Message format



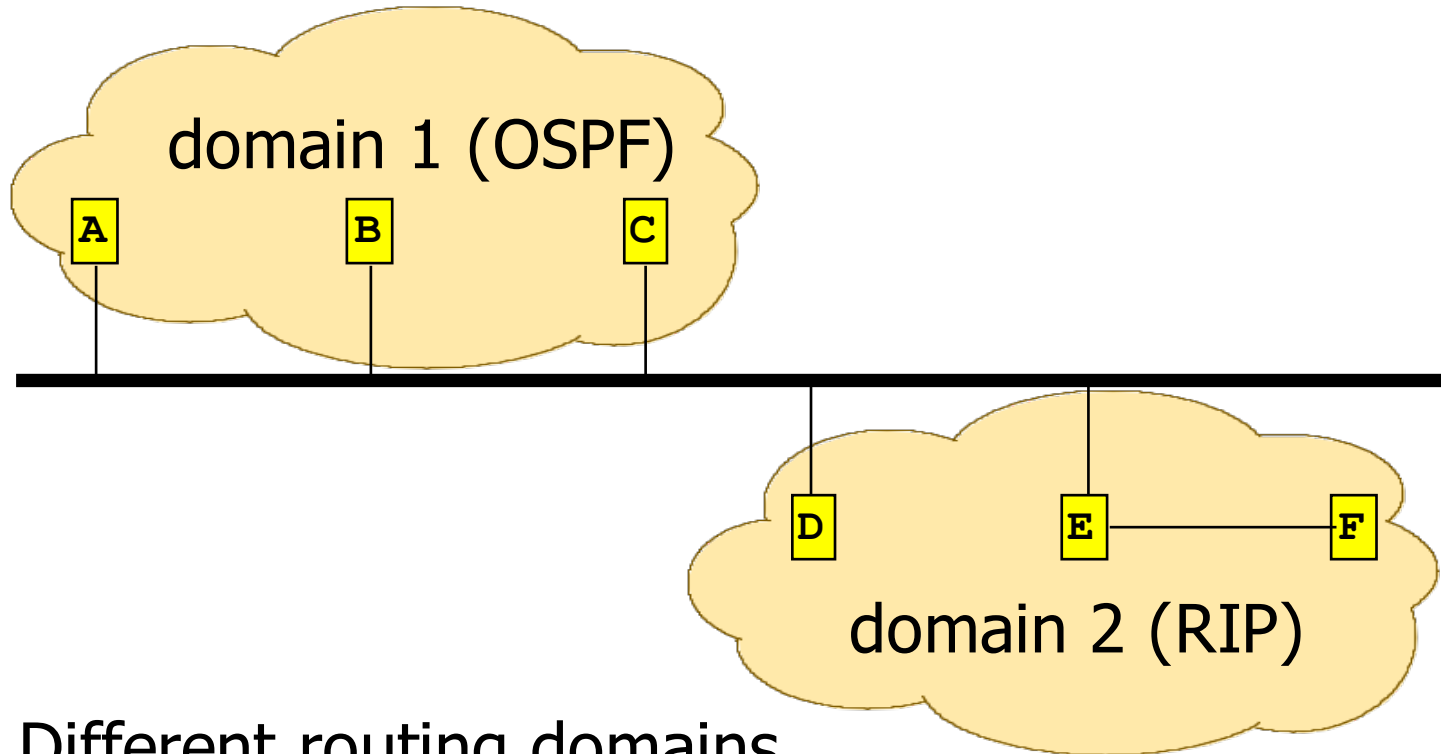
- Command, version unchanged
- One address family - authentication
- Next router
 - used at the border of different routing domains (e.g. RIP and OSPF)
- Route tag
 - for external routes (used by BGP)

Announcing netmasks



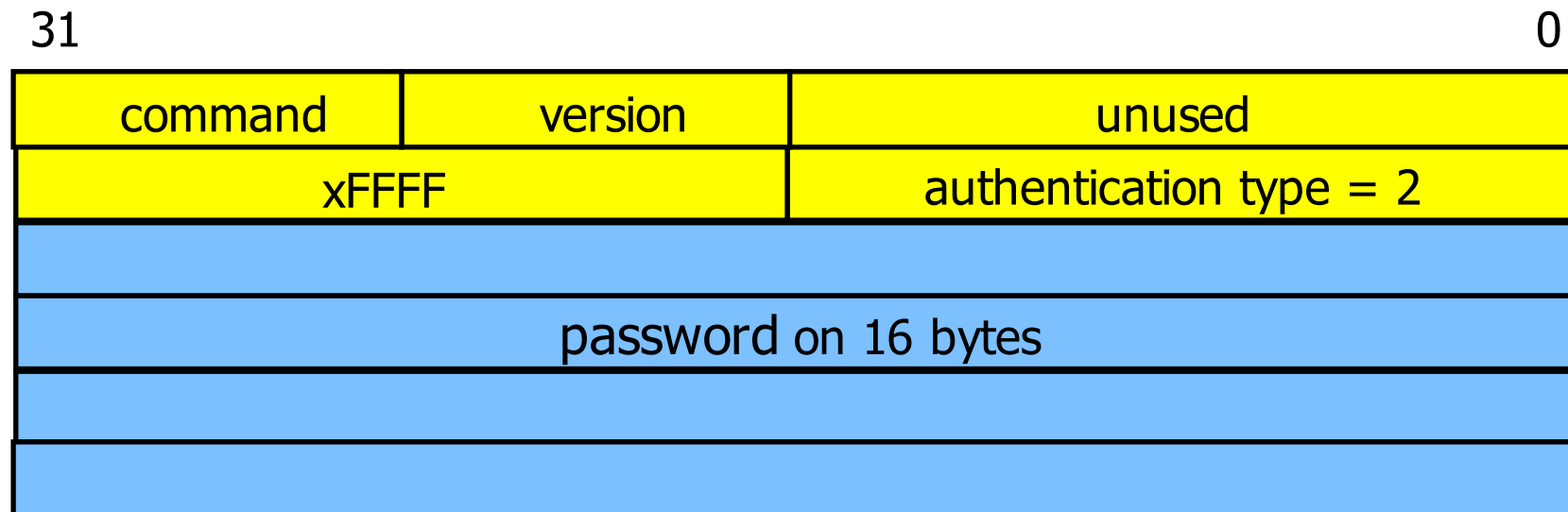
- E can forward to 10.2.0.0
- Packet to 10.2.0.1 can go through F

Routing domains



- Different routing domains
 - e.g. routers under different administrations that run different routing protocols (RIP, OSPF)
- If A wants to send a packet to F, it goes through D and E
- When announcing F, D adds E as **next router**

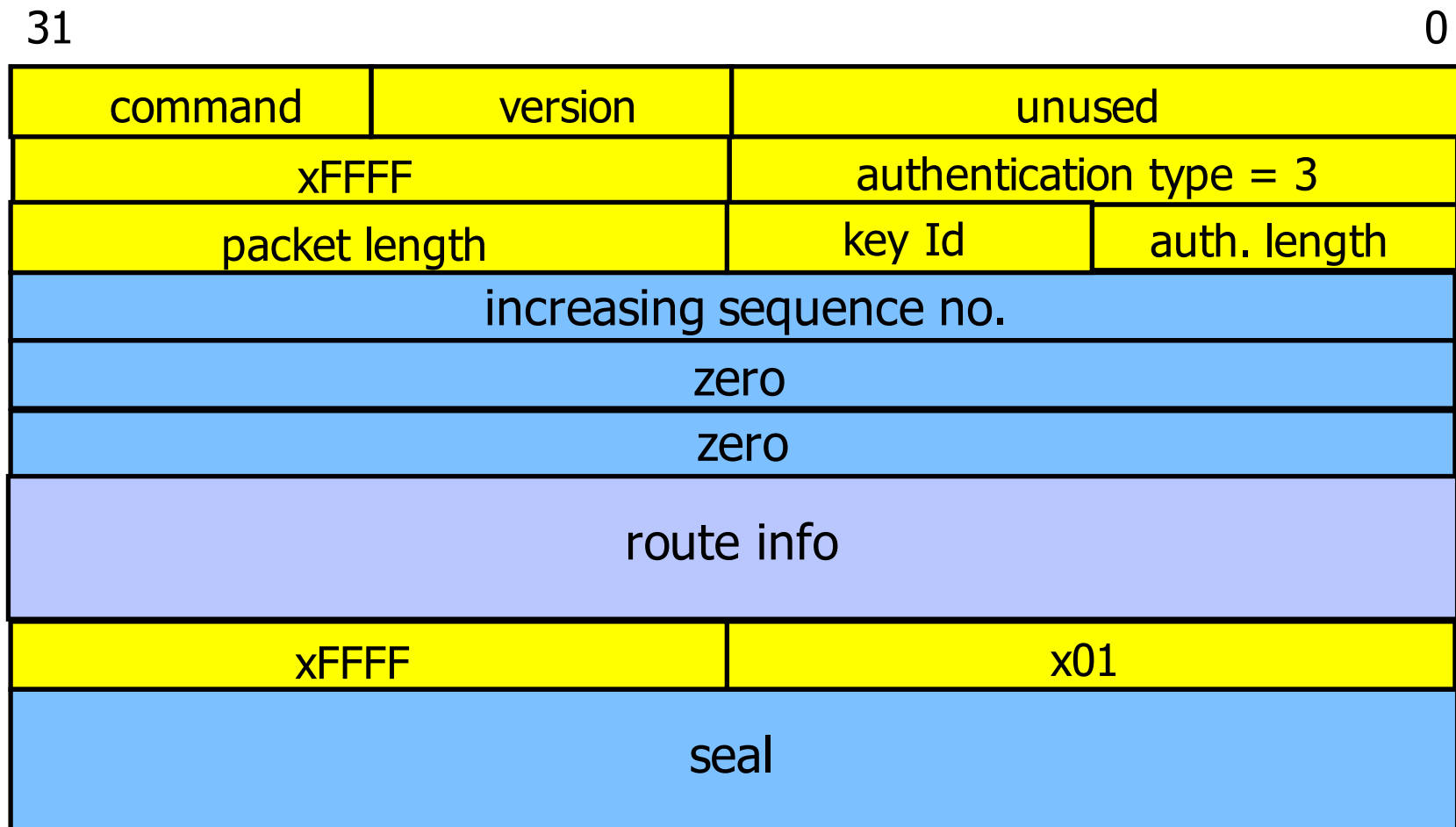
Simple authentication



- Configuration of gated (/etc/gated.conf)

```
rip yes {  
    interface all  
    version 2 multicast  
    authentication simple "qptszwmz"  
}
```


MD5 authentication



MD5 authentication

- Seal
 - MD5 digest on the message using a shared secret
 - sequence number avoids replay attacks
- Configuration of gated (/etc/gated.conf)

```
rip yes {  
    interface all  
    version 2 multicast  
    authentication md5 "qptszwmz"  
}
```

IGRP (Interior Gateway Routing Protocol)

- Proprietary protocol by CISCO
- Metric that estimates the global delay
- Maintains several routes of similar cost
 - load sharing
- Takes into account netmasks
- No limit of 15
 - number of routers included in messages
- Broadcast every 90 sec

Metric example



- Metric

- $\text{Trans} = 10000000/\text{Bandwidth}$ (time to send 10 Kb)
- $\text{delay} = (\text{sum of Delay})/10$
- $m = [K_1 * \text{Trans} + (K_2 * \text{Trans}) / (256 - \text{load}) + K_3 * \text{delay}]$
- default: $K_1=1, K_2=0, K_3=1, K_4=0, K_5=0$
- if $K_5 \neq 0, m = m * [K_5 / (\text{Reliability} + K_4)]$

- Bandwidth in Kb/s, Delay in μs

- At Venus: Route for 172.17/16: $\text{Metric} = 10000000/784 + (20000+1000)/10 = 14855$
- At Saturn: Route for 12./8: $\text{Metric} = 10000000/224 + (20000 + 1000)/10 = 46742$

Conclusion

- Main distance vector protocols
- Largely deployed (Unix BSD `routed`)
- Simplicity
- Slow convergence
- Not suited for large and complex networks
 - Link State protocols should be used instead